

[Date of registration]

[Number of appeal against examiner's decision
of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

(43)公開日 平成14年1月11日(2002.1.11)

(51)Int.Cl. ⁷	識別記号	F I	ページ(参考)
H 0 4 Q 9/00	3 2 1	H 0 4 Q 9/00	3 2 1 E 5 B 0 4 5
	3 0 1		3 0 1 E 5 B 0 8 9
G 0 6 F 9/44	5 3 0	G 0 6 F 9/44	5 3 0 M 5 B 0 9 8
	3 6 0	9/46	3 6 0 D 5 K 0 3 3
13/00	3 5 7	13/00	3 5 7 A 5 K 0 4 8
審査請求 未請求 請求項の数16 O L (全 16 頁) 最終頁に続く			

(33) 優先権主張国 日本 (J P)

東京都品川区北品川6丁目7番35号

東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内

弁理士 松隈 秀盛

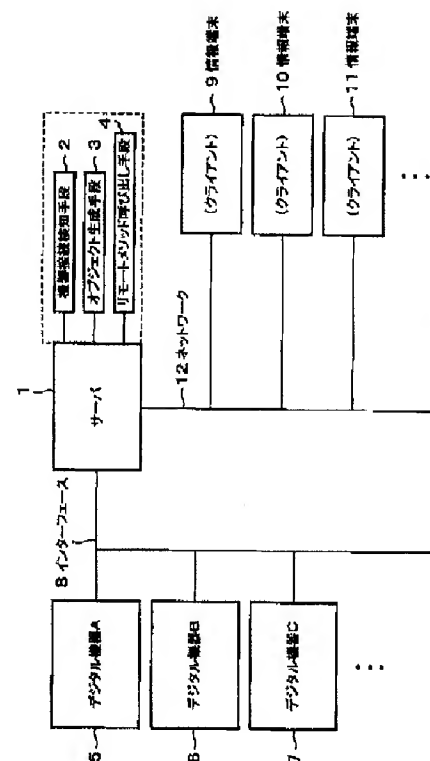
HA02 HA03

(54) 【発明の名称】 機器制御装置および機器制御システム

(57) 【要約】

【課題】 複数の被制御機器に対してネットワーク経由でサーバーから代理制御を行うことができ、また、サーバーに代理オブジェクトの実装を行うことができる機器制御装置および機器制御システムを提供する。

【解決手段】 サーバ１は、複数の被制御機器５～７のそれぞれがインターフェース８を介して接続されたことを検知する機器接続検知手段２と、接続が検知された複数の被制御機器５～７の固有の情報に基づいて、オブジェクト指向のプログラムを用いた制御情報となる代理オブジェクトの自動生成削除を行うためのオブジェクトを自動生成するオブジェクト生成手段３とを備えるものである。



本実座の形態が適用されるサーバを用いた機器制御システムの構成図

【特許請求の範囲】

【請求項 1】 インターフェースを介して接続された複数の被制御機器に対して、各々動作をさせるための制御情報を生成して、供給する機器制御装置において、上記複数の被制御機器のそれぞれが上記インターフェースを介して接続されたことを検知する機器接続検知手段と、

接続が検知された上記複数の被制御機器の固有の情報に基づいて、オブジェクト指向のプログラムを用いた上記制御情報となる代理オブジェクトの自動生成削除を行うためのオブジェクトを自動生成するオブジェクト生成手段と、

を備えたことを特徴とする機器制御装置。

【請求項 2】 請求項 1 記載の機器制御装置において、上記代理オブジェクトは、ネットワークを介して接続された複数の情報端末からアクセス可能であることを特徴とする機器制御装置。

【請求項 3】 請求項 2 記載の機器制御装置において、上記オブジェクトの有する機能であるメソッドを呼び出すリモートメソッド呼び出し手段を備えたことを特徴とする機器制御装置。

【請求項 4】 請求項 2 記載の機器制御装置において、上記被制御機器が削除状態のとき、上記オブジェクトにより生成された対応する代理オブジェクトを削除することを特徴とする機器制御装置。

【請求項 5】 インターフェースを介して接続された複数の被制御機器に対して、各々動作をさせるための制御情報を生成して、供給する機器制御装置において、上記複数の被制御機器のそれぞれが上記インターフェースを介して接続されたことを検知する機器接続検知手段と、

接続が検知された上記複数の被制御機器の固有の情報に基づいて、オブジェクト指向のプログラムを用いた上記制御情報となる代理オブジェクトの自動生成削除を行うためのオブジェクトを自動生成するオブジェクト生成手段と、

上記代理オブジェクトの有する機能であるメソッドにより、上記被制御機器をアクセス不能状態にロックすると共に、上記ロックを解除するアンロックを行うロック手段とを備えたことを特徴とする機器制御装置。

【請求項 6】 請求項 5 記載の機器制御装置において、上記ロック手段は、ネットワークを介して接続された複数の情報端末からアクセス可能であることを特徴とする機器制御装置。

【請求項 7】 請求項 6 記載の機器制御装置において、上記情報端末から上記被制御機器がロックされたとき、所定時間以内に上記情報端末から上記ロックの更新を可能とすることを特徴とする機器制御装置。

【請求項 8】 請求項 6 記載の機器制御装置において、上記情報端末から上記被制御機器がロックされたとき、

所定時間以内に上記情報端末から上記ロックの更新が行われないとき、上記ロックは無効とされることを特徴とする機器制御装置。

【請求項 9】 請求項 6 記載の機器制御装置において、上記被制御機器へのアンロックは、上記情報端末に対する一意情報により行われることを特徴とする機器制御装置。

【請求項 10】 インターフェースを介して接続された複数の被制御機器に対して、各々動作をさせるための制御情報を生成して、供給する機器制御装置において、上記複数の被制御機器のそれぞれが上記インターフェースを介して接続されたことを検知する機器接続検知手段と、

接続が検知された上記複数の被制御機器の固有の情報に基づいて、オブジェクト指向のプログラムを用いた上記制御情報となる代理オブジェクトの自動生成削除を行うためのオブジェクトを自動生成するオブジェクト生成手段と、

上記代理オブジェクトの有する機能であるメソッドにより、対応する上記被制御機器の名前付けをすると共に、上記名前を取得する名前付け手段とを備えたことを特徴とする機器制御装置。

【請求項 11】 請求項 10 記載の機器制御装置において、上記名前付け手段は、ネットワークを介して接続された複数の情報端末からアクセス可能であることを特徴とする機器制御装置。

【請求項 12】 請求項 10 記載の機器制御装置において、上記複数の被制御機器の固有の情報を記憶する記憶手段を備えたことを特徴とする機器制御装置。

【請求項 13】 請求項 12 記載の機器制御装置において、上記記憶手段は不揮発性であることを特徴とする機器制御装置。

【請求項 14】 請求項 12 記載の機器制御装置において、上記名前付け手段により、上記被制御機器の名前付けを行う際に、上記代理オブジェクトの有する機能であるメソッドに対応する上記被制御機器の名前が上記記憶手段に登録されることを特徴とする機器制御装置。

【請求項 15】 請求項 12 記載の機器制御装置において、上記名前付け手段により、上記被制御機器の名前の取得を行う際に、上記代理オブジェクトの有する機能であるメソッドに対応する上記被制御機器の名前が上記記憶手段から取得されることを特徴とする機器制御装置。

【請求項 16】 各々がインターフェースを介して接続され、各々固有の情報を記憶する記憶手段を備え、制御情報により各々動作する複数の被制御機器と、

上記複数の被制御機器のそれぞれが上記インターフェースを介して接続されたことを検知する機器接続検知手段と、接続が検知された上記複数の被制御機器の固有の情報に基づいて、オブジェクト指向のプログラムを用いた上記制御情報となる代理オブジェクトの自動生成削除を行うためのオブジェクトを自動生成するオブジェクト生成手段と、を有する機器制御装置と、
上記機器制御装置にネットワークを介して接続され、上記代理オブジェクトに対してアクセス可能である複数の情報端末と、
を備えた機器制御システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、例えば、インターフェースを介して接続された複数の被制御機器をネットワーク経由のアクセスにより制御する機器制御装置および機器制御システムに関するものである。

【0002】

【従来の技術】従来、インターフェースを介して接続された複数の被制御機器に対して、各々動作をさせるための制御情報を生成して、供給する機器制御装置を用いる制御システムがあった。

【0003】また、複数の被制御機器に対してネットワーク経由でサーバーから情報を供給するネットワークシステムがあった。

【0004】

【発明が解決しようとする課題】しかし、上述した従来の制御システムおよびネットワークシステムでは、サーバーにネットワークを介して情報端末を接続し、情報端末からサーバーにアクセスすることにより、機器制御装置に替えて、サーバーに被制御機器の制御を代理させて行わせることが要求された場合に、サーバーがこの代理制御を行うための機能を有していないため、サーバーが被制御機器の代理制御を行うことができないという不都合があった。

【0005】また、特に、オブジェクト指向のプログラムを用いた被制御機器に対する制御を行うための代理オブジェクトをサーバーに実装することが要求された場合に、サーバーがこの代理オブジェクトを実装するための機能を有していないため、サーバーに代理オブジェクトの実装を行うことができないという不都合があった。

【0006】そこで、本発明は、かかる点に鑑みてなされたものであり、複数の被制御機器に対してネットワーク経由でサーバーから代理制御を行うことができ、また、サーバーに代理オブジェクトの実装を行うことができる機器制御装置および機器制御システムを提供することを課題とする。

【0007】

【課題を解決するための手段】本発明の機器制御装置は、インターフェースを介して接続された複数の被制御

機器に対して、各々動作をさせるための制御情報を生成して、供給する機器制御装置において適用される。

【0008】本発明の機器制御装置は、複数の被制御機器のそれぞれがインターフェースを介して接続されたことを検知する機器接続検知手段と、接続が検知された複数の被制御機器の固有の情報に基づいて、オブジェクト指向のプログラムを用いた制御情報となる代理オブジェクトの自動生成削除を行うためのオブジェクトを自動生成するオブジェクト生成手段とを備えるものである。

10 【0009】また、本発明の機器制御装置は、代理オブジェクトの有する機能であるメソッドにより、被制御機器をアクセス不能状態にロックすると共に、ロックを解除するアンロックを行うロック手段を備えるものである。

【0010】また、本発明の機器制御装置は、代理オブジェクトの有する機能であるメソッドにより、対応する上記被制御機器の名前付けをすると共に、上記名前を取得する名前付け手段を備えるものである。

20 【0011】また、本発明の機器制御システムは、各々がインターフェースを介して接続され、各々固有の情報を記憶する記憶手段を備え、制御情報により各々動作する複数の被制御機器と、複数の被制御機器のそれぞれがインターフェースを介して接続されたことを検知する機器接続検知手段と、接続が検知された複数の被制御機器の固有の情報に基づいて、オブジェクト指向のプログラムを用いた制御情報となる代理オブジェクトの自動生成削除を行うためのオブジェクトを自動生成するオブジェクト生成手段と、を有する機器制御装置と、機器制御装置にネットワークを介して接続され、代理オブジェクトに対してアクセス可能である複数の情報端末とを備えたものである。

【0012】従って本発明によれば、以下の作用をする。まず、被制御機器の接続の検知の処理が行われる。機器接続検知手段は、複数の被制御機器のそれぞれがインターフェースを介して接続されたことを検知する。

【0013】次に、接続された被制御機器に対応するオブジェクトの生成の処理が行われる。オブジェクト生成手段は、接続が検知された複数の被制御機器の固有の情報に基づいて、オブジェクト指向のプログラムを用いた制御情報となる代理オブジェクトの自動生成削除を行うためのオブジェクトを自動生成する。ここで、機器の所定の記憶手段に、機器固有の情報が書かれている。

【0014】ここで、オブジェクト生成手段は、インターフェースを介して接続される被制御機器の情報を調査し、それに対応した適当なオブジェクトを生成する。

【0015】そして、オブジェクトのサービススタートメソッドの呼び出しの処理が行われる。サービススタートメソッドは、被制御機器の固有情報に基づいて代理オブジェクトを生成し、ネットワーク上のクライアントの情報端末からのアクセスを可能にする。

【0016】また、被制御機器の削除の検知の処理が行われる。機器接続検知手段は、複数の被制御機器のそれぞれがインターフェースから削除されたことを検知する。

【0017】そして、オブジェクトのサービスストップメソッドの呼び出しの処理が行われる。サービスストップメソッドは、サービススタートメソッドで生成した代理オブジェクトへのネットワーク上のクライアントの情報端末からのアクセスを不可にし、代理オブジェクトを削除する。

【0018】さらに、接続された被制御機器に対応するオブジェクトの削除の処理が行われる。オブジェクト生成手段は、接続が検知された複数の被制御機器の固有の情報に基づいて、オブジェクト指向のプログラムを用いた制御情報となる代理オブジェクトの自動生成削除を行うためのオブジェクトを自動削除する。

【0019】ここで、オブジェクト生成手段は、インターフェースから削除される被制御機器の情報を調査し、それに対応した適当な代理オブジェクトを削除する。

【0020】

【発明の実施の形態】本実施の形態の機器制御装置は、サーバに接続されたデジタル機器をサーバ上の代理オブジェクト経由でネットワーク上からアクセス可能とするものであり、代理オブジェクトの自動生成または削除、機器のロック、および機器への永続的な名前付けを行う機能を有するものである。

【0021】〔機器制御システム〕図1は、本実施の形態が適用されるサーバを用いた機器制御システムの構成を示す図である。この機器制御システムは、サーバ1と、複数の被制御機器となるデジタル機器5〜7と、デジタル機器5〜7をサーバ1に接続するインターフェース8と、複数のクライアントとなる情報端末9〜11と、情報端末9〜11をサーバ1に接続するネットワーク12とを有して構成される。

【0022】サーバ1は、複数の被制御機器5〜7のそれぞれがインターフェース8を介して接続されたことを検知する機器接続検知手段2と、接続が検知された複数の被制御機器5〜7の固有の情報に基づいて、オブジェクト指向のプログラムを用いた制御情報となる代理オブジェクトの自動生成削除を行うためのオブジェクトを自動生成するオブジェクト生成手段3と、情報端末9〜11から、オブジェクトの有する機能であるメソッドを呼び出すリモートメソッド呼び出し手段4を有して構成される。

【0023】このように構成されたこの機器制御システムは、以下のような動作をする。サーバ1において、機器接続検知手段2は、複数の被制御機器5〜7のそれぞれがインターフェース8を介して接続されたことを検知する。

【0024】また、オブジェクト生成手段3は、接続が

検知された複数の被制御機器5〜7の固有の情報に基づいて、オブジェクト指向のプログラムを用いた制御情報となる代理オブジェクトの自動生成削除を行うためのオブジェクトを自動生成する。

【0025】これにより、サーバ1にデジタル機器5〜7が接続されたときに、デジタル機器5〜7をネットワーク12経由で情報端末9〜11から操作するための代理オブジェクトが動作可能な状態となるように生成される。

10 【0026】また、サーバ1には複数のクライアントの情報端末9〜11が接続されているので、代理オブジェクトは、ネットワーク12を介して接続された複数の情報端末9〜11からアクセス可能な状態である。

【0027】そこで、リモートメソッド呼び出し手段4は、ネットワーク12を介して接続された複数の情報端末9〜11からのアクセスにより、オブジェクトの有する機能であるメソッドを呼び出す。

20 【0028】これにより、呼び出されたメソッドを用いて各デジタル機器5〜9が動作される。このように、サーバ1において、代理オブジェクトを用いることにより、あたかもネットワーク12を介して接続された複数の情報端末9〜11から直接デジタル機器5〜9を操作して動作をさせるようにできる。

【0029】〔代理オブジェクトの自動生成削除〕上述した図1に示した機器制御システムでは、サーバ1にサーバプロセスと呼ばれる動作プログラムが常駐していて、このサーバプロセスにより上述した機器接続検知手段2と、オブジェクト生成手段3と、リモートメソッド呼び出し手段4の動作が行われる。ここでは、代理オブジェクトの生成動作を中心に説明する。

【0030】図2は、サーバの動作を示すフローチャートである。この動作は、サーバ1のサーバプロセスにより行われるものである。図2において、ステップS1でデジタル機器の接続の検知が行われる。具体的には、機器接続検知手段2は、複数の被制御機器5〜7のそれぞれがインターフェース8を介して接続されたことを検知する。

40 【0031】ステップS2で、接続されたデジタル機器に対応するオブジェクト(AvcUnit)の生成が行われる。具体的には、オブジェクト生成手段3は、接続が検知された複数の被制御機器5〜7の固有の情報に基づいて、サーバプロセスにより、オブジェクト指向のプログラムを用いた制御情報となる代理オブジェクトの自動生成削除を行うためのオブジェクト(AvcUnit)を自動生成する。例えば、IEEE1394フォーマットでは、機器のコンフィギュレーションROM(ConfigurationROM(Read Only Memory))に、機器固有の情報が書かれている。

【0032】ここで、オブジェクト生成手段3は、インターフェース8を介して接続されるデジタル機器の情報

を調査し、それに対応した適当なオブジェクト(AvcUnit)を生成する。

【0033】ステップS3で、オブジェクト(AvcUnit)のサービススタート(serviceStart)メソッドの呼び出しが行われる。サービススタートメソッドは、被制御機器5~7の固有情報に基づいて代理オブジェクトを生成し、ネットワーク12上のクライアントの情報端末9~11からのアクセスを可能にする。

【0034】ステップS4でデジタル機器の削除の検知が行われる。具体的には、機器接続検知手段2は、複数の被制御機器5~7のそれぞれがインターフェース8から削除されたことを検知する。

【0035】ステップS5で、オブジェクト(AvcUnit)のサービスストップ(serviceStop)メソッドの呼び出しが行われる。サービスストップメソッドは、代理オブジェクトへのネットワーク12上のクライアントの情報端末9~11からのアクセスを終了状態にし、代理オブジェクトを削除する。

【0036】ステップS6で、接続されたデジタル機器に対応するオブジェクト(AvcUnit)の削除が行われる。

【0037】ここで、オブジェクト生成手段3は、インターフェース8から削除されるデジタル機器の情報を調査し、それに対応した適当なオブジェクト(AvcUnit)を削除する。

【0038】図3は、オブジェクト(AvcUnit)のサービススタート(serviceStart)メソッドの動作を示すフローチャートである。例えば、インターフェース8がIEEE1394フォーマットに準拠したものであり、デジタル機器5~9がAV/C(エー・ブイ・シー)コマンド(AV/C Digital Interface Command Set)で制御可能なものである場合に、デジタル機器には機器の機能を示すサブユニット(Subunit)タイプという情報がある。

【0039】ここでは、ユニットはデジタル機器そのもののことで、サブユニットは、デジタル機器の機能を司るものである。従って、サブユニットの組み合わせがユニットとなる。ユニットの中をどのような機能単位で分けるかは、適宜決められる。

【0040】ステップS11で、機器情報の調査が行われる。具体的には、ここでは、デジタル機器のサブユニット(Subunit)タイプの調査が行われる。

【0041】ステップS12で、サブユニットタイプNNに対応した代理オブジェクト(SubunitTypeNN_impl)の生成が行われる。具体的には、サブユニットタイプ(SubunitType)には、0以上の整数NNが割り当てられている。

【0042】ここで、代理オブジェクトをサブユニット

タイプ別に、SubunitTypeNN_implのように命名することを取り決めることにより、サブユニットタイプに応じた代理オブジェクトをオブジェクト(AvcUnit)から自動生成することができる。

【0043】ステップS13で、代理オブジェクト(SubunitTypeNN_impl)をネットワークから利用可能にする。具体的には、リモートメソッド呼び出し手段4は、ネットワーク12を介して接続された複数の情報端末9~11からのアクセスにより、代理オブジェクト(SubunitTypeNN_impl)を利用可能にする。

【0044】例えば、デジタル機器として、テレビジョンユニットはチューナサブユニット(受信機能)と、モニタサブユニットとの組み合わせが考えられ、また、テレビ一体型ビデオテープレコーダは、チューナサブユニット(受信機能)と、モニタサブユニットと、テープレコーダ/プレイヤーサブユニット(記録機能/再生機能)との組み合わせが考えられる。このように機能単位として適当なサブユニットが定められる。

【0045】なお、上述したサブユニットは、仮想的な機能単位であり、実際の回路構成と一致するとは限らないものである。また、例えば、回路の中で、デコーダブロックのようにAV/Cコマンドによるコントロールの必要がないブロックのように、どのサブユニットにも入らないものがある。

【0046】図4は、オブジェクト(AvcUnit)のサービスストップ(serviceStop)メソッドの動作を示すフローチャートである。

【0047】ステップS21で、代理オブジェクト(SubunitTypeNN_impl)をネットワークから利用不可能にする。具体的には、リモートメソッド呼び出し手段4は、ネットワーク12を介して接続された複数の情報端末9~11からのアクセスにより、代理オブジェクト(SubunitTypeNN_impl)を利用不可能にする。

【0048】ステップS22で、代理オブジェクト(SubunitTypeNN_impl)の削除が行われる。

【0049】このようにして、オブジェクト(AvcUnit)のサービスストップ(serviceStop)メソッドは、サービススタート(serviceStart)メソッドで自動生成した代理オブジェクト(SubunitTypeNN_impl)をネットワークから利用不可能にして、削除する。

【0050】なお、サービススタート(serviceStart)メソッドは、オブジェクト(AvcUnit)が生成された直後に、サーバプロセスにより呼び出されて、サービスストップ(serviceStop)メソッドは、デジタル機器が利用不可能になったときに、サーバプロセスにより呼び出される。

【0051】 [代理オブジェクトによる機器のロック]
図5は、プロキシ (代理) オブジェクトによる機器のロック (lock) メソッドの動作を示すフローチャートである。

【0052】 ステップS31で、クライアント (A) の情報端末からロック (lock) メソッドの呼び出しが行われる。具体的には、リモートメソッド呼び出し手段4によりプロキシ (代理) オブジェクトによる機器のロック (lock) メソッドが呼び出される。

【0053】 ステップS32でAではないクライアントの情報端末からロック (lock) メソッドが取得済み
10 可否かを判断する。

【0054】 ステップS32でAではないクライアントの情報端末からロック (lock) メソッドが取得済みでないときは、ステップS34へ進んで、クライアント (A) の情報端末がロック (lock) に成功する。

【0055】 ステップS32でAではないクライアントの情報端末からロック (lock) メソッドが取得済みのときは、ステップS33へ進んで、クライアント
(A) の情報端末がロック (lock) に失敗する。

【0056】 そして、ロック (lock) 更新時には、再度、ステップS31～ステップS34までの処理および判断を繰り返す。

【0057】 このようにして、サーバ1上においてアクセス対象のデジタル機器5～7に対して生成される代理オブジェクトは、ネットワーク上の複数のクライアントの情報端末からアクセス可能である。

【0058】 ここで、上述したように、クライアントの情報端末によってデジタル機器をロックして排他的にデジタル機器を操作したい場合がある。例えば、デジタル
30 機器のリモートコントローラをクライアントの情報端末で実現したいという要求がある場合には以下のようなことが想定される。

【0059】 このような要求から、代理オブジェクトに上述したロックメソッドを実現するロック機能を実装し、このロック機能を用いて、デジタル機器の排他的制御を実現することができる。

【0060】 代理オブジェクトは、上述したlock (ロックを得る) メソッドと、後述するunlock (ロックを放棄する) メソッドとを実現する機能を有し
40 ている。これらの機能は、クライアントの情報端末からネットワーク経由で利用することができる。

【0061】 ステップS34で、クライアント (A) の情報端末がロック (lock)、またはロック (lock) 更新に成功すると、ステップS35へ進んで、ロック (lock) 時に指定したN秒の経過を待つ。

【0062】 そして、ステップS36でクライアント (A) の情報端末のロックを無効にする。

【0063】 このように、クライアントの情報端末がロックに成功すると、その後、N秒 (Nは任意の自然数) 50

以内であれば、ロックの更新を行うことができる。なお、このとき、ロックの更新は必ず成功する。

【0064】 ここで、クライアントの情報端末がN秒以内にロックの更新を行わない場合には、ロックは無効になり、別のクライアントの情報端末もロックを行うことができる状態になる。

【0065】 これにより、クライアントの情報端末がロックを放棄しないで長時間放置した場合などのデッドロック状態を回避することができる。

【0066】 図6は、プロキシ (代理) オブジェクトによる機器のアンロック (unlock) メソッドの動作を示すフローチャートである。

【0067】 ステップS41で、クライアント (A) の情報端末からアンロック (unlock) メソッドの呼び出しが行われる。具体的には、リモートメソッド呼び出し手段4によりプロキシ (代理) オブジェクトによる機器のアンロック (unlock) メソッドが呼び出される。

【0068】 ステップS42でAではないクライアントの情報端末からロックが取得済み可否かを判断する。

【0069】 ステップS42でAではないクライアントの情報端末からロックが取得済みでないときは、ステップS44へ進んで、Aのクライアントの情報端末からロックが取得済み可否かを判断する。

【0070】 ステップS44で、Aのクライアントの情報端末からロックが取得済みであるときは、ステップS46へ進んで、クライアント (A) の情報端末がアンロック (unlock) に成功する。

【0071】 ステップS42でAではないクライアントの情報端末からロックが取得済みのときは、ステップS43へ進んで、クライアント (A) の情報端末がアンロック (unlock) に失敗する。

【0072】 ステップS44で、Aのクライアントの情報端末からロックが取得済みでないときは、ステップS45へ進んで、クライアント (A) の情報端末がアンロック (unlock) に失敗する。

【0073】 ここで、アンロック (unlock) は、クライアントの情報端末から一意情報 (例えば、IPアドレス等) を用いて行う。

【0074】 具体的には、代理オブジェクトは、現在ロックを取得しているクライアントの情報端末の一意情報を保持していて、この状態でアンロックの要求があると、アンロックを要求したクライアントの情報端末の一意情報と、ロックを取得しているクライアントの情報端末の一意情報とを比較する。

【0075】 この一意情報が同一であれば、アンロックは成功し、そうでなければ、アンロックは失敗する。

【0076】 [代理オブジェクトによる機器の永続的な名前付け] 図7は、プロキシ (代理) オブジェクトによる機器のセットニックネーム (setNicknam

e) メソッドの動作を示すフローチャートである。ステップS51で、クライアントの情報端末からセットニックネーム (setNickname) メソッドの呼び出しが行われる。具体的には、リモートメソッド呼び出し手段4によりプロキシ (代理) オブジェクトによる機器の永続的な名前付けを示すセットニックネーム (setNickname) メソッドが呼び出される。

【0077】ステップS52でプロキシ (代理) オブジェクトに対応する機器用のサーバ上のデータベースを検出する。具体的には、サーバ1は、デジタル機器5〜7それぞれに固有の情報に対応したファイルシステム等の不揮発性のデータベースを有している。このデータベースには、各デジタル機器に対応する名前を格納する。この名前は、クライアントが情報端末9〜11からの入力により命名することができ、これは、クライアントがネットワーク上のデジタル機器を識別する際に用いられる。

【0078】名前の例としては、「A宅リビングのビデオデッキ」、「B宅書斎のCS (Communication Satellite) チューナ」等が考えられる。このサーバ上のデータベースをクライアントの情報端末が利用する際の手続を以下に説明する。

【0079】図2に示したサーバの動作において生成された代理オブジェクトは、ネットワーク上の複数のクライアントの情報端末から利用可能である。

【0080】代理オブジェクトは、setNickname (機器の名前を付ける) のメソッドを実現する機能を有している。これらの機能は、クライアントの情報端末からネットワーク経由で利用することができる。

【0081】そこで、ステップS53でデータベースをロックし、他のクライアントの情報端末からの書き込みを不可にする。

【0082】ステップS54でデータベースにセットニックネーム (setNickname) メソッドで指定された機器の新しい名前を登録する。具体的には、クライアントの情報端末からセットニックネーム (setNickname) メソッドにより、機器への名前付けを行う。この際、サーバ上のデータベースには代理オブジェクトに対応する機器の新しい名前が登録される。ここで、このデータベースは不揮発性なので、新しく登録された名前情報は代理オブジェクトが削除された後にも、永続的に保存される。

【0083】ステップS55でデータベースのロックを解除する。

【0084】図8は、プロキシ (代理) オブジェクトによる機器のゲットニックネーム (getNickname) メソッドの動作を示すフローチャートである。

【0085】ステップS61で、クライアントの情報端末からゲットニックネーム (getNickname) メソッドの呼び出しが行われる。具体的には、リモート

メソッド呼び出し手段4によりプロキシ (代理) オブジェクトによる機器の永続的な名前の取得を示すゲットニックネーム (getNickname) メソッドが呼び出される。

【0086】ステップS62でプロキシ (代理) オブジェクトに対応する機器用のサーバ上のデータベースを検出する。具体的には、サーバは、デジタル機器それぞれに固有の情報に対応したファイルシステム等の不揮発性のデータベースを有している。このデータベースには、各デジタル機器に対応する名前を格納する。この名前は、クライアントの情報端末9〜11からの入力により命名することができ、これは、クライアントがネットワーク上のデジタル機器を識別する際に用いられる。

【0087】代理オブジェクトは、getNickname (機器の名前を得る) のメソッドを実現する機能を有している。これらの機能は、クライアントの情報端末からネットワーク経由で利用することができる。

【0088】そこで、ステップS63でデータベースをロックし、他のクライアントの情報端末からの書き込みを不可にする。

【0089】ステップS64でデータベースにある機器の名前を得るゲットニックネーム (getNickname) メソッドで指定された機器の新しい名前を得る。具体的には、クライアントの情報端末からゲットニックネーム (getNickname) メソッドにより、データベースに登録されている機器の名前の取得を行う。この際、サーバ上のデータベースに登録されている代理オブジェクトに対応する機器の名前を得る。

【0090】ステップS65でデータベースのロックを解除する。ステップS66で機器の名前を返す。

【0091】[オブジェクト指向のプログラミング] オブジェクト指向のプログラミング言語として適用される、Java (サンマイクロシステムズ社の商標) について説明する。

【0092】まず、クラスについて説明する。一般的にオブジェクトは「データ」と「動作」を含んだものである。Javaにおいてオブジェクトに相当するものがクラスである。従って、1つのクラスが1つのオブジェクトになる。

【0093】クラスはJavaの基本単位である。プログラムの実行も1つのクラスを呼び出すことで実行される。クラスでは、データを変数、動作をメソッドと呼ぶ。1つのクラスは、変数とメソッドから成り立っている。この2つの要素をクラスのなかに書き込むことによって、クラスの性質と振る舞いが決まり、オブジェクトとして利用できる。

【0094】次に、変数およびメソッドについて説明する。クラスの性質を定義するのは変数である。変数は、いくつかの基本的な型を持ち、クラスのなかで値を入れる「器」になる。この器に値を入れることでより具体的

なクラスとなる。メソッドはクラスに含まれる変数やほかのクラスを利用してプログラムの動作を定義する。

【0095】例えば、ビデオテープレコーダでは、その動作を示すものとして、「記録」、「再生」、「停止」、「早送り」、「巻き戻し」などのメソッドを持っている。クラスがJava VM（仮想マシン）に呼び出されて、1つのオブジェクトとして振る舞いをする状態を示す、クラスがインスタンス化されたときには、このメソッドを利用することができる。

【0096】クラスは機能を引き継いだ子供のクラスを作ることができる。これを継承という。このとき子供のクラスは親のクラスの変数やメソッドを引き継ぐ。これにより、プログラムを作成する際にすべての動作を記述する必要がなくなる。

【0097】次に、RMI（Remote Method Invocation）について説明する。RMIはリモートメソッド呼び出し手段4に対応する、アプリケーション作成を支援するクラスとしてのAPI（Application Programming Interface）である。RMIはデータをやりとりするだけでなく、メソッドやオブジェクトをやりとりすることが可能である。C言語のリモートプロシージャコール（RPC）に似たものである。

【0098】RMIを使うと、ネットワークを介して複数のコンピュータにまたがって処理を行うプログラムを作成することができる。これを分散コンピューティングという。RMIはJava独自の規格のため、Javaプログラム同士では簡単に通信することができるが、ほかのプログラム言語で作成されたプログラムなどとは通信することはできない。

【0099】【IEEE1394シリアルバス】本実施の形態のインターフェース8に適用されるIEEE1394シリアルバス（以下、1394シリアルバスという。）の概要を説明する。まず、接続形態を説明する。1394シリアルバスでは接続形態が限定されていて、各機器をループのないツリー状に接続することで、最大63台の機器を1つのバスに接続することができる。各機器のポートは、受信したデータ信号を連続的に他のポートへ伝送することにより、データ信号がバス全体に伝搬する。

【0100】次に、ケーブルについて説明する。2組の差動信号線TPAおよびTPBと、電源ペアVG（グラウンド）およびVP（電源）で構成される。TPAおよびTPBの2本の信号線で、バスの動的なコンフィギュレーション、バスの使用権を取得するためのアービトラージョン、データ信号の伝搬を行う。TPAには常時バイアスをかけており、TPBではそれを検出することにより、アクティブなケーブルの接続の有無を判定している。

【0101】次に、物理レイヤー部の信号について説明

する。TPA、TPBの2本の信号線は、「1」、「0」、「Z」の3値をとる。「Z」は接続されたポートのどちらにも駆動されていないハイインピーダンス状態を意味する。TPAからはストロブと呼ばれる信号を、TPBからはデータを送信する。受信側ではデータとストロブの排他的論理和をとることでクロックを得て、クロックの変化点でデータを読み取る。

【0102】次に、バス初期化について説明する。各ポートは、接続相手のTPAが出力するバイアスの有無を検出して、ノードが接続されたか取り外されたかを判定する。ポートの接続状態の変化を検出したノードは、他の接続のあるポートに対して一定時間バスリセット信号を送信する。これを受信したノードは、さらに接続のある他のポートにバスリセット信号を送信する。これを繰り返し、最終的にバスに接続された全ノードにバスリセット信号が伝わる。バスリセット信号を受信したノードは、それ以前の形態情報や、各自のノードIDをクリアする。その後、各ノードは自らが、バスに接続された各ノードが隣接する複数のノードに接続されている状態のブランチか、または隣接するノードは1つだけのリーフかを認識するようにし、さらに、各ノードの親子関係付けを行い、ツリー構造におけるルートノードの決定が行われる。

【0103】次に、アドレスについて説明する。64ビット幅のアドレス空間のうち、上位10ビット（最大1023）のバスIDとそれに続く6ビット（最大（63））のノードIDで機器を識別する。残りの48ビットはノード内のアドレス空間として割り当てられる。また、このアドレス空間の所定のレジスタ空間には、CSR（Control and Status Register）と、CFR（Configuration Register、コンフィギュレーションROMと同じ）が標準化される。

【0104】次に、非同期通信（アシンクロナス通信）について説明する。1394インターフェースで用いるデータパケットの転送方法の一つとして、非同期通信がある。これは片方向のデータパケット転送である。送信側はデータパケット転送先アドレスをパケットヘッダーに明記し、バスに送信する。データパケットはバス上のすべてのノードまで伝搬する。パケットヘッダーに明記された転送先アドレスに対応するノードは、そのデータパケットを受信し、受信結果（ack）を返送する。その一連の転送プロセスをアシンクロナスサブアクションと呼ぶ。

【0105】ここで、アシンクロナスサブアクションを開始するためには、サブアクションギャップと呼ばれる一定期間、バスがアイドル状態にならなくてはならない。また、受信側がデータパケットを受け取って、ackを返送する間もバスはアイドル状態となり、この間隔をackギャップと呼ぶ。ackギャップはサブアクション

ョンギャップに比べて充分短いため、他のサブアクションが開始されることはない。

【0106】次に、アイソクロナス通信について説明する。1394 インターフェースで用いるデータパケットの転送方法のもう一つの方法として、アイソクロナス通信がある。アイソクロナス通信は、バス上に1台存在するサイクルマスターが一定間隔で送信するサイクルスタートパケットに同期して行われる。アイソクロナス通信の送信側のノードはサイクルスタートパケットを受信するとアイソクロナスギャップを待ってアービトレーションを開始し、アイソクロナスパケットを送信する。他にも送信ノードが存在する場合には、引き続きアイソクロナスギャップで、アービトレーションを開始し、アイソクロナスパケットを送信する。

【0107】ここで、アイソクロナスギャップは、サブアクションギャップより充分短い期間になっているため、この間にアシンクロナス通信を行いたいノードがいても、サブアクションギャップが検出できないため送信できない。つまり、毎サイクル、アイソクロナスパケットを送信するノードに優先権が与えられることになる。また、アイソクロナス通信は、アシンクロナス通信のように、データ転送先のアドレスを指定することなく、バスにブロードキャストされる。アイソクロナスパケットには0〜63までのチャンネル番号が割り振られ、ノードは必要なチャンネル番号のアイソクロナスパケットを受信すればよい。

【0108】また、上述した本実施の形態では、オブジェクト指向のプログラミング言語として、Java を使用する例を示したが、これに限らず、他のC言語を使用するようにしても良い。

【0109】また、上述した本実施の形態では、インターフェースとしてIEEE1394フォーマットを使用する例を示したが、これに限らず、他のUSB(Universal Serial Bus)を使用するようにしても良い。

【0110】また、上述した本実施の形態では、機器制御装置として、サーバ1を使用する例を示したが、これに限らず、同様の構成および作用効果を奏するものであれば、セットトップボックス、ルータを使用するようにしても良い。

【0111】また、上述した本実施の形態では、ネットワーク12としては、インターネットを使用する例を示したが、これに限らず、LAN(Local Area Network)を使用するようにしても良い。

【0112】また、上述した本実施の形態では、クライアントとしては、パーソナルコンピュータなどの情報端末9〜11を使用する例を示したが、これに限らず、携帯電話を使用するようにしても良い。

【0113】

【発明の効果】この発明の機器制御装置は、複数の被制

御機器のそれぞれがインターフェースを介して接続されたことを検知する機器接続検知手段と、接続が検知された複数の被制御機器の固有の情報に基づいて、オブジェクト指向のプログラムを用いた制御情報となる代理オブジェクトの自動生成削除を行うためのオブジェクトを自動生成するオブジェクト生成手段とを備えるので、接続された複数の被制御機器を代理オブジェクトを用いてネットワークを経由して操作する場合に、機器の挿抜に応じて代理オブジェクトの自動生成削除を実現できると共に、代理オブジェクトの実装を簡略化することができるという効果を奏する。

【0114】また、この発明の機器制御装置は、代理オブジェクトの有する機能であるメソッドにより、被制御機器をアクセス不能状態にロックすると共に、ロックを解除するアンロックを行うロック手段を備えるので、接続された複数の被制御機器を代理オブジェクトを用いてネットワークを経由して操作する場合に、排他制御を行うために機器のロックまたは解除を実現できると共に、代理オブジェクトの実装を簡略化することができるという効果を奏する。

【0115】また、この発明の機器制御装置は、代理オブジェクトの有する機能であるメソッドにより、対応する上記被制御機器の名前付けをすると共に、上記名前を取得する名前付け手段を備えるので、接続された複数の被制御機器を代理オブジェクトを用いてネットワークを経由して操作する場合に、繰り返し制御を行うために機器の永続的な名前付けを実現できると共に、代理オブジェクトの実装を簡略化することができるという効果を奏する。

【0116】また、この発明の機器制御システムは、各々がインターフェースを介して接続され、各々固有の情報を記憶する記憶手段を備え、制御情報により各々動作する複数の被制御機器と、複数の被制御機器のそれぞれがインターフェースを介して接続されたことを検知する機器接続検知手段と、接続が検知された複数の被制御機器の固有の情報に基づいて、オブジェクト指向のプログラムを用いた制御情報となる代理オブジェクトの自動生成削除を行うためのオブジェクトを自動生成するオブジェクト生成手段と、を有する機器制御装置と、機器制御装置にネットワークを介して接続され、代理オブジェクトに対してアクセス可能である複数の情報端末とを備えたので、機器制御装置に接続された複数の被制御機器を代理オブジェクトを用いてネットワークを経由して操作する場合に、機器の挿抜に応じて代理オブジェクトの自動生成削除を実現できると共に、代理オブジェクトの実装を簡略化するシステムを構築することができるという効果を奏する。

【図面の簡単な説明】

【図1】本実施の形態が適用されるサーバを用いた機器制御システムの構成を示す図である。

【図 2】サーバの動作を示すフローチャートである。

【図 3】オブジェクト (AvcUnit) のサービススタート (serviceStart) メソッドの動作を示すフローチャートである。

【図 4】オブジェクト (AvcUnit) のサービスストップ (serviceStop) メソッドの動作を示すフローチャートである。

【図 5】代理オブジェクトによる機器のロック (lock) メソッドの動作を示すフローチャートである。

【図 6】代理オブジェクトによる機器のアンロック (unlock) メソッドの動作を示すフローチャートである。

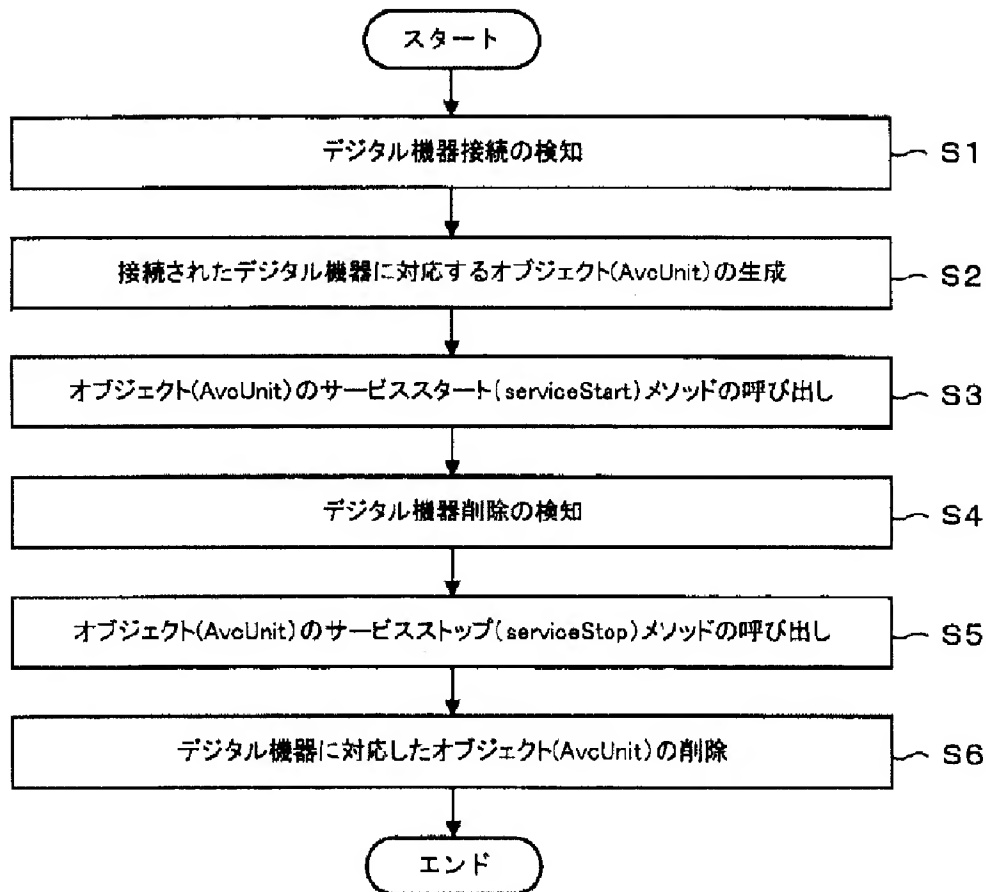
【図 7】代理オブジェクトによる機器のセットニックネーム (setNickname) メソッドの動作を示すフローチャートである。

【図 8】代理オブジェクトによる機器のゲットニックネーム (getNickname) メソッドの動作を示すフローチャートである。

【符号の説明】

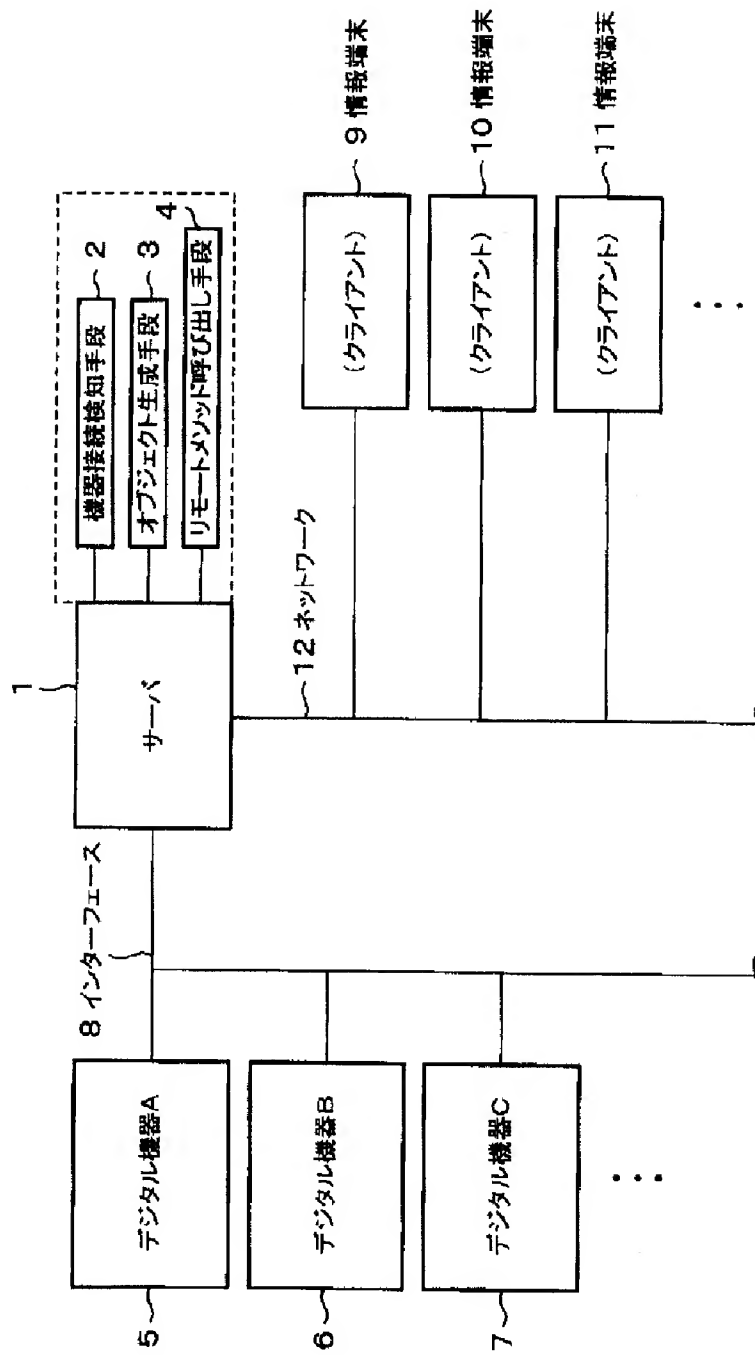
1……サーバ、2……機器接続検知手段、3……オブジェクト生成手段、4……リモートメソッド呼び出し手段、5～7……デジタル機器、8……ネットワーク、9～11……情報端末、12……ネットワーク

【図 2】



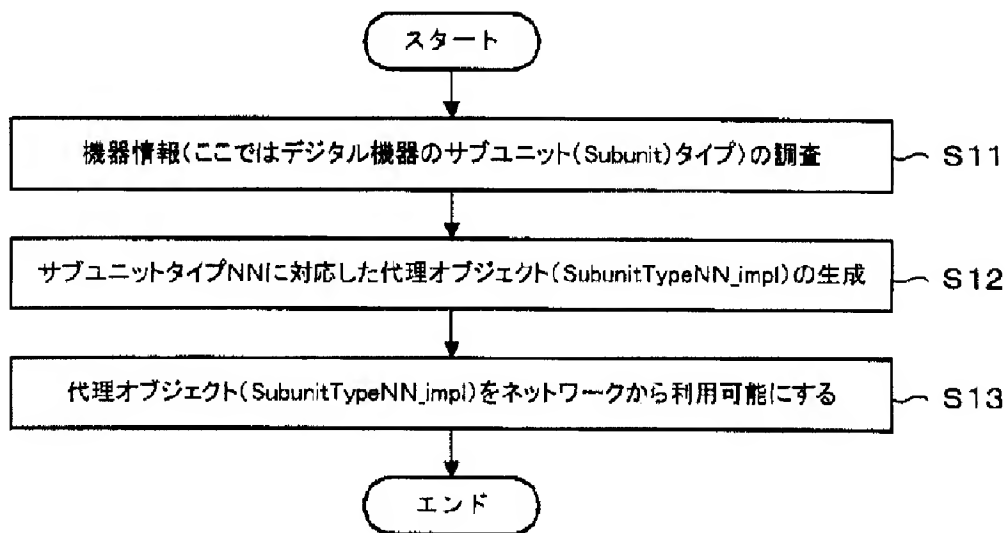
サーバの動作を示すフローチャート

【図 1】



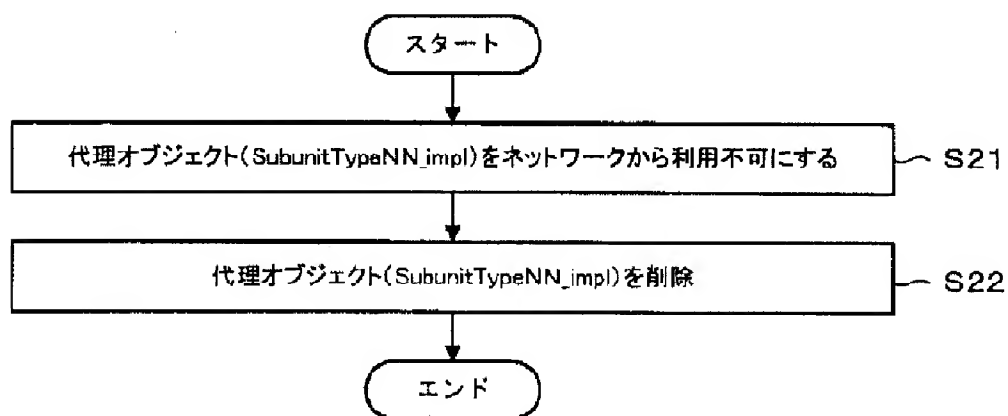
本実施の形態が適用されるサーバを用いた機器制御システムの構成図

【図 3】



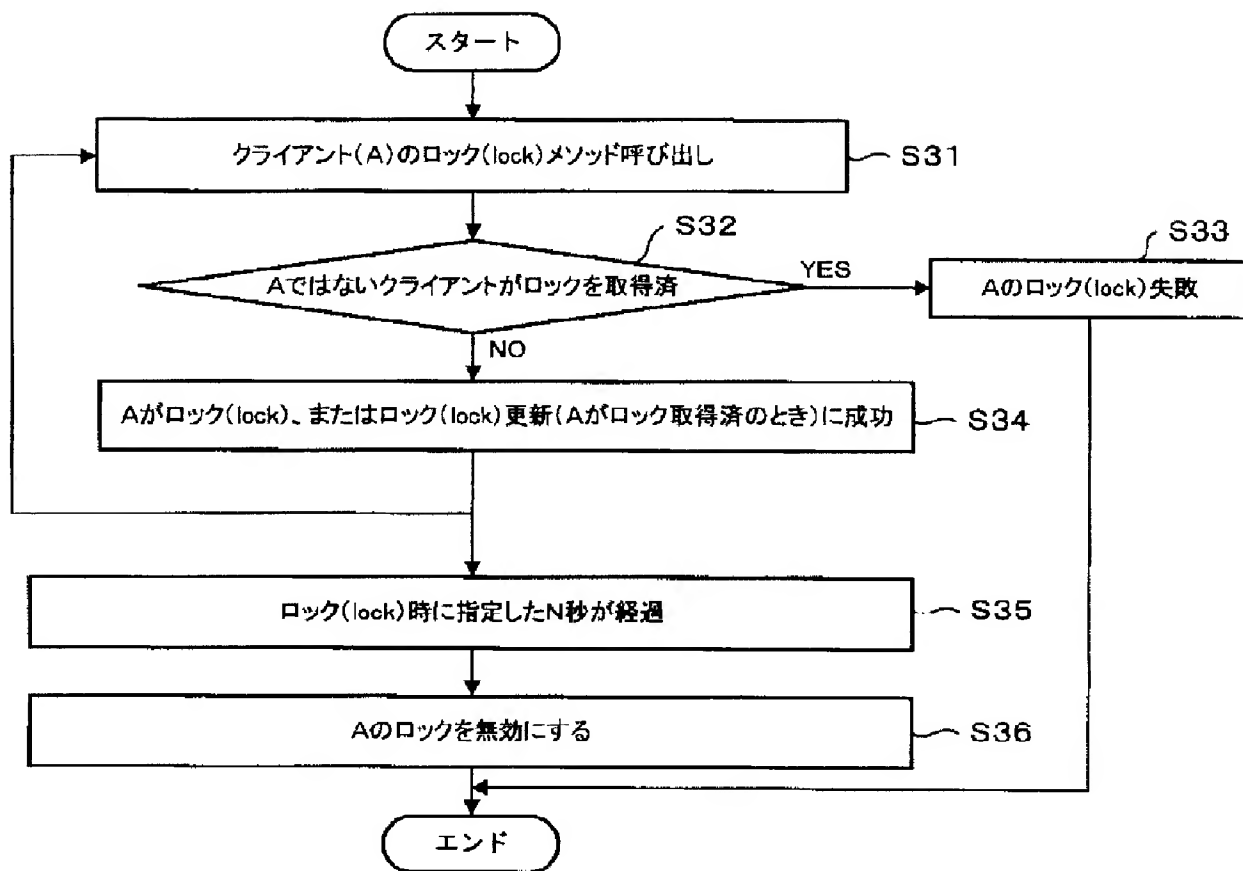
オブジェクト(AvcUnit)のサービススタート(serviceStart)
メソッドの動作を示すフローチャート

【図 4】



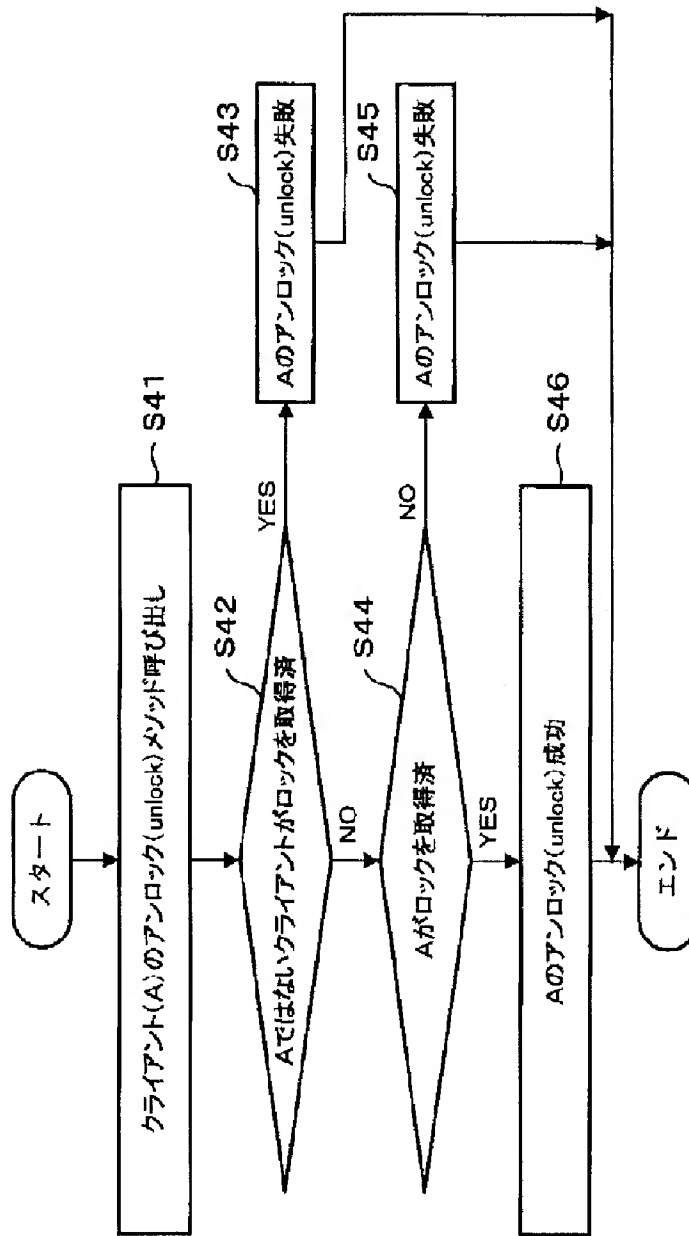
オブジェクト(AvcUnit)のサービスストップ(serviceStop)
メソッドの動作を示すフローチャート

【図 5】



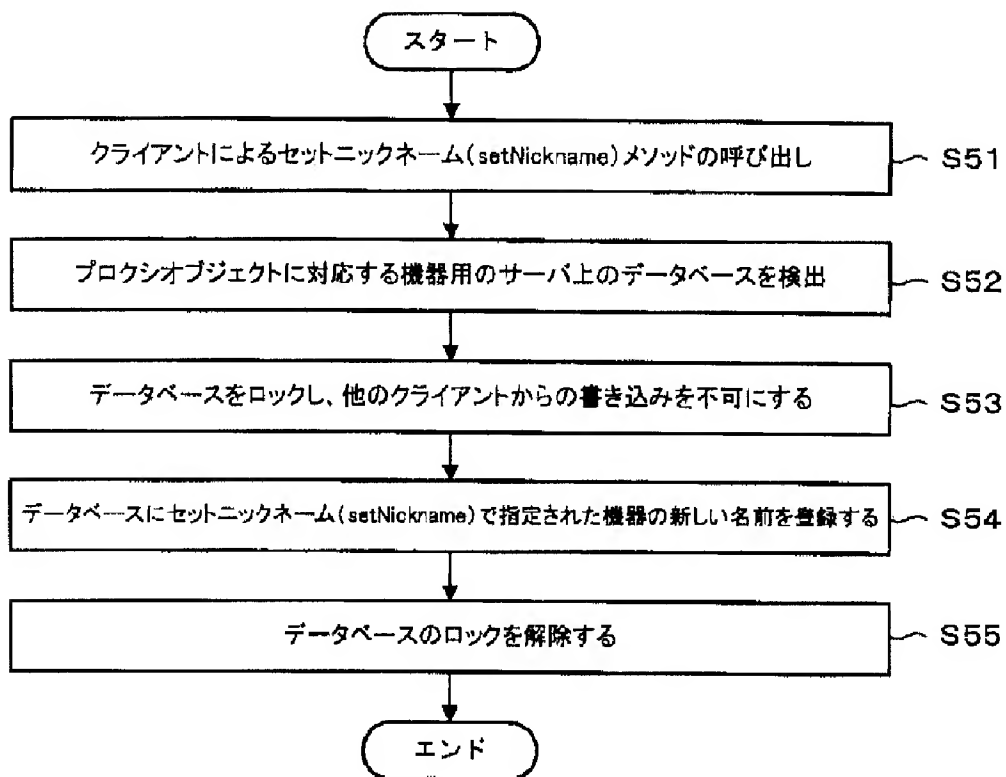
プロキシオブジェクトによる機器のロック(lock)メソッドの動作を示すフローチャート

【図 6】



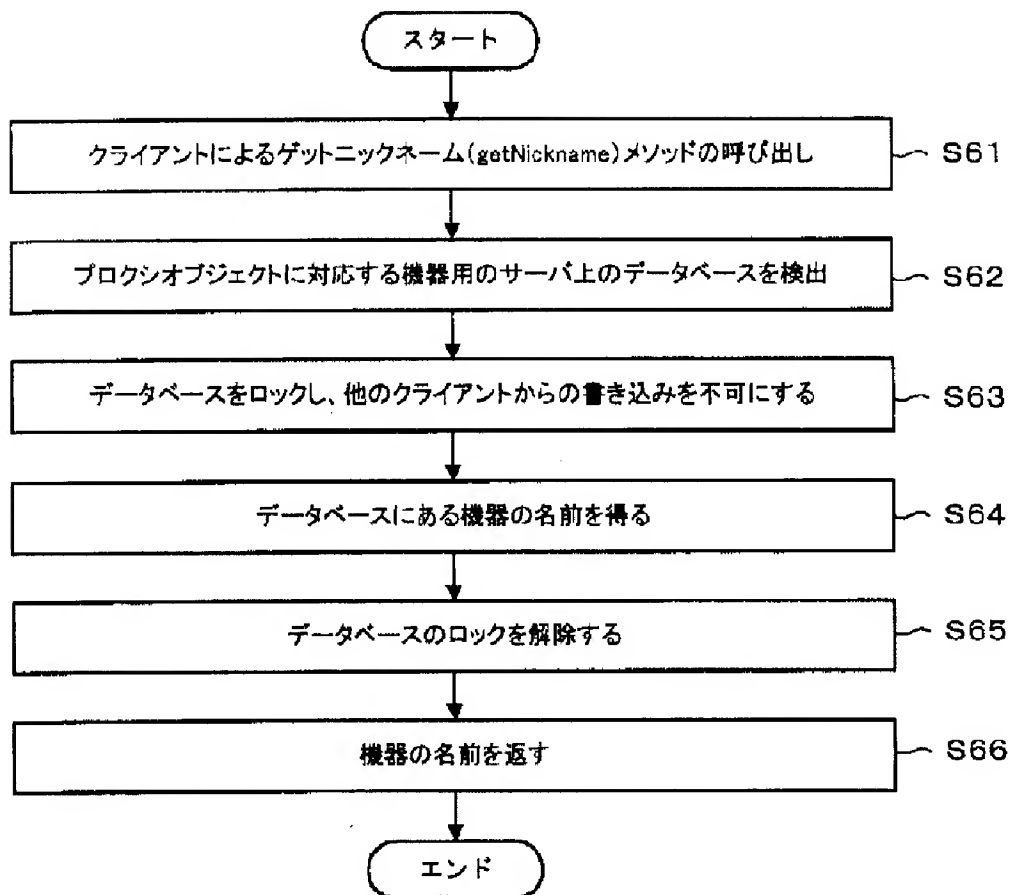
プロキシオブジェクトによる機器のアンロック(unlock)メソッドの動作を示すフローチャート

【図 7】



プロキシオブジェクトによる機器のセットニックネーム
(setNickname)メソッドの動作を示すフローチャート

【図 8】



プロキシオブジェクトによる機器のゲットニックネーム
(getNickname)メソッドの動作を示すフローチャート

フロントページの続き

(51) Int. Cl.⁷

G 0 6 F 15/16

H 0 4 L 12/28

識別記号

6 2 0

F I

G 0 6 F 15/16

H 0 4 L 11/00

テーマコード(参考)

6 2 0 W

3 1 0 D